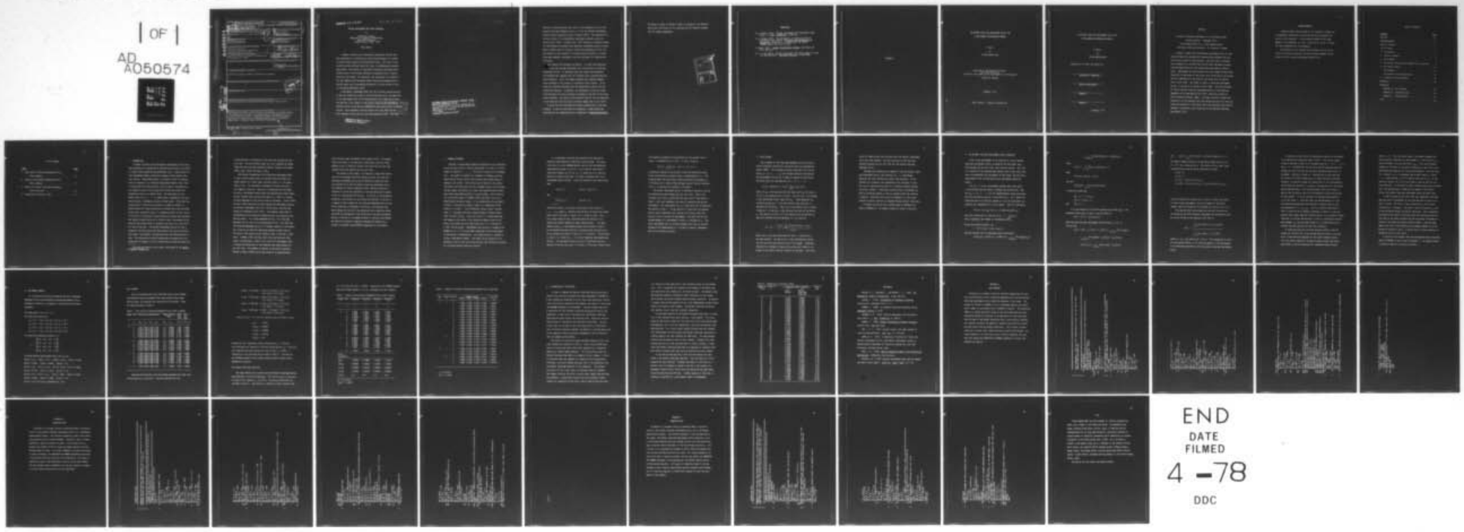
AD-A050 574 TEXAS A AND M UNIV COLLEGE STATION DEPT OF INDUSTRIA--ETC F/G 15/5
OPTIMAL REPLACEMENT FOR SHOCK PROCESSES.(U)
1977 R M FELDMAN AFOSR-76-3111
UNCLASSIFIED AFOSR-TR-78-0038 NL

| OF |

AD
A050574



END
DATE
FILMED
4 -78
DDC

AD A050574

DDC FILE COPY

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFOSR-TR-78-0038	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): OPTIMAL REPLACEMENT FOR SHOCK PROCESSES	5. TYPE OF REPORT & PERIOD COVERED: Final rept.
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Richard M./Feldman	8. CONTRACT OR GRANT NUMBER(s): AFOSR-76-3111
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Texas A&M University, Industrial Engineering Department, College Station, TX 77843	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 61102F 2304/A5 A5
11. CONTROLLING OFFICE NAME AND ADDRESS: Air Force Office of Scientific Research/NM, Bolling AFB, DC 20332	12. REPORT DATE: 1977 51p.
	13. NUMBER OF PAGES: 38
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

optimal replacement, Semi-Markov Processes, Markov renewal theory

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The research performed under this grant successfully developed a method for determining an optimal state-age dependent replacement policy for a semi-Markov shock model. Comparisons between the method developed here and the standard policy iteration approach show a significant improvement in computer speed and memory requirements.

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

403 969.

JOB

AFOSR-TR- 78-0038

AFOSR-76-3111

OPTIMAL REPLACEMENT FOR SHOCK PROCESSES

by

Richard M. Feldman
Department of Industrial Engineering
Texas A&M University
College Station, Texas 77843

Final Report

A number of models and corresponding replacement policies have been developed for stochastically deteriorating systems in an effort to reduce system operation and maintenance costs. The focus of this project has been directed toward the class of semi-Markovian replacement models. Such models are semi-Markov stochastic processes where a given state of the process represents a specified level of deterioration of the system. By convention, the state space is a subset of the real numbers with increasing numbers denoting increasing deterioration levels and an increasing probability of total failure as well as increasing maintenance costs.

A semi-Markov replacement model has the following characteristics: 1) when the system has a jump in its deterioration level, the magnitude of the jump depends only on the deterioration level immediately before the jump and 2) the length of time between jumps may governed by an arbitrary distribution function dependent on the current level of deterioration. The replacement policies studied with this model are the state dependent policy and the state-age dependent policy. The usual

method for obtaining either the optimal state dependent policy or the optimal state-age dependent policy is to use the standard semi-Markov decision theory approach as given by Howard (1971). The application of decision theory to the semi-Markov replacement problem is given in detail by Kao (1973). Feldman (1976, 1977) develops an alternate method for determining the optimal state dependent replacement policy by using Markov renewal theory to obtain a closed form expression for the cost. The purpose of this research is to extend these procedures to include state-age dependent replacement rules and determine its computational efficiency.

The research was extremely successful. A closed form expression for the long term average replacement cost was derived for state-age replacement policy. An algorithm using the closed form expression was developed and compared with the standard policy iteration-decision theory approach. Forty five sample problems were compared ranging from a problem of four states to a problem of thirty states. In all cases the algorithm developed here was significantly faster than the traditional approach. In general, the advantage of using the closed form algorithm over policy iteration increased as the size of the state space increased. The ratio of the execution time for the new algorithm to the execution time for policy iteration ranged from 1.61 to 18.70.

A proof that the algorithm will always converge has not yet been finished. As soon as the proofs are completed, a paper giving the algorithm and the comparisons will be submitted to <u>Operations Research</u>.

The Master's thesis of Tilden N. Mikel is included in the appendix which gives the details of the algorithm and the computer program used for making comparisons.

REFERENCES

R. M. Feldman (1976). Optimal replacement with semi-Markov shock models. J. Appl. Probability, 13:108-117.

R. M. Feldman (1977). The maintenance of systems governed by semi-Markov shock models. The Theory and Applications of Reliability, edited by C. P. Tsokos and I. N. Shimi, Academic Press, New York, 215-226.

R. Howard (1971). Dynamic Probabilistic Systems, John Wiley and Sons, New York.

E. P. C. Kao (1973). Optimal replacement rules when changes of state are semi-Markovian. Operations Research, 21:1231-1249.

APPENDIX

AN OPTIMAL STATE-AGE REPLACEMENT POLICY FOR
A SEMI-MARKOV DETERIORATION PROCESS

A Thesis

by

TILDEN NEWTON MIKEL

Submitted to the Graduate College of
Texas A&M University
in partial fulfillment of the requirement for the degree of

MASTER OF SCIENCE

December, 1977

Major Subject: Industrial Engineering

AN OPTIMAL STATE-AGE REPLACEMENT POLICY FOR

A SEMI-MARKOV DETERIORATION PROCESS

A Thesis

by

TILDEN NEWTON MIKEL

Approved as to style and content by:

____
(Chairman of Committee)

____
(Head of Department)

____
(Member)

____
(Member)

December, 1977

ABSTRACT

An optimal state-age replacement for a semi-Markov deterioration process. (December 1977)

Tilden Newton Mikel, B.S., Texas A&M University

Chairman of Advisory Committee: Dr. Richard M. Feldman

A number of models and corresponding replacement policies have been developed for stochastically deteriorating systems with both military and industrial applications. One such class of problems can be modeled as a discrete time finite state semi-Markov process with the deterioration of the system being described by a Markov chain. Replacement policies possible for such a model include those based only on the state of the system, only on the age of the system, and on a combination of the state of the system and its sojourn time in that state. The latter of these, a state-age replacement policy, is the policy of concern to this paper. The only procedure to find an optimal state-age replacement policy is the State-Age Dependent Policy developed by Kao (1972), requiring solution by Policy Iteration (Howard, 1960). This paper derives a closed form expression for the expected long term average cost per unit time and, using the properties of the Markov chain and additional constraints, develops a systematic search technique for the Optimal State-Age Replacement Policy.

ACKNOWLEDGEMENTS

I extend my most sincere thanks to Dr. Richard M. Feldman for his eagerness, enthusiasm, and untiring patience throughout the course of this research. I also extend my thanks to the other members of my committee, Dr. Guy L. Curry and Dr. Larry J. Ringer, for their cooperation in this endeavor.

The funding for this research was provided by the Air Force Office of Scientific Research, Air Force Systems Command, United States Air Force, under Grant Number AFOSR-76-3111.

TABLE OF CONTENTS

LIST OF TABLES

1. INTRODUCTION

A number of models and corresponding replacement policies have been developed for stochastically deteriorating systems in an effort to reduce system operation and maintenance costs. Applications for such replacement models and policies abound in both industry and the military. Among these models is a specific class of problems that are referred to as "Markovian Replacement Models" (Kao, 1972). In these models the deterioration of the system is represented by the change in state of the system which follows the transition-probability matrix of a Markov chain. The state space of this Markov chain is $\{0, 1, \ldots, L\}$, where state 0 represents a new system and state L represents a totally failed system. Given that the system starts in state 0 and no action is taken to replace it, deterioration will cause transitions to successively less desirable states until eventually state L is reached and total failure occurs. If the costs of replacing a failed system are greater than replacing the system at some time prior to failure, then a replacement policy based on the current state of occupancy may result in a lower overall cost to the user. The optimal replacement policy for such a system will be that policy that best balances the costs associated with repair, replacement, and operation over some defined period of time. This balance may be easily approached by assuming that the system will be needed in service indefinitely and that the objective

---

The style and format of this thesis follow that of the Journal of Applied Probability.

of the problem is minimization of the long term average cost per unit time. This basic Markov model was first suggested by Derman (1962) and has since been studied by Barlow, Proschan, and Hunter (1965), Ross (1970), and Taylor (1975).

One obvious disadvantage of the Markovian Replacement Model is that a Markov process does not consider that the longer the system remains in a given state, the more likely it is to deteriorate or fail. An increasing likelihood of failure with age is very common in practice. Age may be incorporated into the model by using a semi-Markov process where the deterioration among states follows a Markov chain and the time spent in a state is a random variable dependent on the current state of occupancy. Çinlar (1975) and Ross (1970) describe the theory behind the semi-Markov process, and optimal replacement rules for a finite state, discrete time system are developed by Kao (1972). Kao develops three types of replacement policies based on information that is available to the decision maker: (1) a State Dependent Policy, (2) a State-Age Dependent Policy, and (3) an Age Dependent Policy. Kao shows that the State-Age Dependent policy is in general superior to the others, but it may not be worth the additional expense to keep track of the state of the system and the length of time it has been in that state. Feldman (1976) considers both finite and infinite state spaces, and develops a control limit policy for replacement using a closed form expression for the expected long term average cost per unit time. This method is superior to the Policy Iteration Method of Howard (1960) and has been shown to be computationally

more efficient than the method of Kao (Lampe, 1977). The computational efficiency is of particular significance since all three methods arrive at identical control limit policies with the same expected long term average cost per unit time.

The purpose of this paper is to develop a closed form expression for the expected long term average cost per unit time for a state-age replacement policy and to develop an algorithm using this expression to find the optimal state-age replacement policy. This method will then be compared to Kao's method using policy iteration to verify the optimality of the new method and to demonstrate its computational efficiency. In Section 2, the parameters, conditions, and notation of the problem are defined; in Section 3, a summary of Kao's method for the State-Age Dependent Policy using Policy Iteration is presented; in Section 4, the expected long term average cost per unit time expression and a selective search algorithm are developed to find the optimal state-age replacement policy; in Section 5, an example problem is solved by each method; and Section 6 presents a comparison between the methods for a variety of problems using FORTRAN programming for each method.

2. PROBLEM STATEMENT

Consider a system whose underlying condition can be identified at any distinct point in time by classifying it in one of a finite number of states 0, 1, ..., L. This set of states will be denoted by E. The system is subject to a sequence of randomly occuring deterioration forces, each of which causes some random amount of damage to the system. These forces may be shocks to the system, the effects of normal wear and tear, abnormal stress on the system, fatigue, power fluctuations or surges, and numerous other factors that could cause deterioration in a given system. This narrative will follow the format of Feldman (1976) and refer to all such deterioration forces as shocks. A new system prior to any deterioration is classified as being in state 0; a completely failed system is in state L. The intermediate states, state 1 through state L-1, represent discrete, ordered degrees of deterioration short of total failure. Since each shock is of random magnitude and damage to the system is the cumulative damage caused by the shocks, the system can only deteriorate until it is replaced or a total failure occurs. Replacement upon failure is assumed to be mandatory, as it is of no economic advantage to delay replacement if the system is nonproductive. This deterioration is assumed to follow a semi-Markov process. The shocks to the system will be assumed to occur at the end of the discrete time intervals according to a discrete holding time mass function.

It is necessary to define the notation to be used and to establish some additional conditions on the problem. The transition matrix for the imbedded Markov chain of the semi-Markovian deterioration process is an (L+1) x (L+1) matrix consisting of individual elements $p(i,j)$ for $i,j \in E$, where $p(i,j)$ is the probability of moving from state i to state j assuming there is no replacement before failure. Since the system can only deteriorate, then

$$p(i,j) = 0 \qquad \text{for } j \leq i \text{ and } i \neq L$$

and

$$\sum_{j=0}^{L} p(i,j) = 1 \qquad \text{for all } i \in E.$$

The policy vector $\underline{\kappa}$ is the set of decisions at each state (κ_0, κ_1, ..., κ_L), where κ_i indicates the decision to replace the system given that it has been in state i for κ_i units of time. Each κ_i has a lower bound of zero, indicating that the decision is to replace the system immediately upon reaching state i. The upper bound of each κ_i is the maximum sojourn time allowed in state i by the holding time mass function, indicating that the decision is to not replace the system as long as it is in state i. The decision at state L is limited to $\kappa_L = 0$ to force immediate replacement upon failure. The semi-Markov kernel $Q(i,j,t)$ is defined as the probability of moving from state i to state j in the time interval $[0,t]$.

The cumulative probability distribution for the sojourn time in state i is denoted by $H(i,\cdot)$ for $i \in E$ and is given by

$$H(i,t) = \sum_{j \in E} Q(i,j,t) \quad \text{for } i,j \in E,\ t \geq 0.$$

A constraint imposed on the system is that the cumulative probability distribution of sojourn times is nondecreasing in i. This property means that the sojourn times stochastically decrease as i increases and it implies that the mean sojourn times are decreasing in i, a condition Kao imposes in his model.

The optimal replacement policy is defined as the policy giving the minimum long term average cost per unit time. There is an occupancy cost rate a_i associated with each unit of time spent in state i; this cost represents the costs of operation and routine maintenance. The replacement cost c_i is the total cost of replacing the system in state i. It may consist of a base cost for replacement b_i and an additional cost rate per unit time d_i while the system is out of service for replacement. The mean time for the accomplishment of a replacement in state i is defined as $\bar{\tau}_i$. The total replacement cost c_i and the occupancy cost rate a_i are both assumed to be nondecreasing in i in order to insure a reasonable form for the optimal solution.

3. KAO'S METHOD

Kao's method for the State-Age Dependent Policy utilizes a search technique called Policy Iteration which was developed by Howard (1960). This procedure involves selection of an initial policy $\underline{\kappa} = (\kappa_0, \kappa_1, \ldots, \kappa_L)$ and through successive evaluations and improvements the policy is made to converge to the optimal. Initially given are the equations for i = 0, 1, ..., L:

$$v(i,\underline{\kappa}) + g(\underline{\kappa})\bar{\tau}(i,\underline{\kappa}) = f(i,\underline{\kappa}) + \sum_{j=0}^{L} q(i,j,\underline{\kappa})\; v(j)$$

where $\bar{\tau}(i,\underline{\kappa})$ is the mean waiting time under policy $\underline{\kappa}$ for state i, $f(i,\underline{\kappa})$ is the expected cost of state i, and $q(i,j,\underline{\kappa})$ is an element of the semi-Markov kernel under policy $\underline{\kappa}$. These equations are solved by letting $v(L) = 0$ and the relative cost rate $g(\underline{\kappa})$ is obtained for the policy. Thus $v(i)$ is the relative cost of state i when $v(L) = 0$ and $g(\underline{\kappa})$ is the relative cost rate of the policy $\underline{\kappa}$. The relative values of $v(i)$ are used to find the decision κ_i that will minimize the test quantity $\Gamma(i, \kappa_i)$ given by

$$\Gamma(i, \kappa_i) = \frac{f(i,\underline{\kappa}) + \sum_{j=0}^{L} q(i,j,\underline{\kappa})\; v(j,\underline{\kappa}) - v(i,\underline{\kappa})}{\tau(i,\underline{\kappa})} .$$

When a new κ_i has been determined for each i, a new policy $\underline{\kappa}$ has been defined. The new policy is then evaluated by solving for the cost rate and relative costs of the states. Iterations continue in an attempt to improve the policy until there is no change in the policy from one iteration to the next. This repli-

cation of identical policies indicates that the optimal replacement policy has been reached. The cost rate $g(\underline{\kappa})$ is the long term expected average cost per unit time for the optimal State-Age Dependent Policy.

Although this procedure will generally find the optimal state-age replacement policy, the solutions to L + 1 simultaneous equations for each iteration are quite time consuming. As the problem size increases, the computations also increase, as does the size of required array space for a computer software package using Kao's method. If multiple solutions exist, the method can take an inordinate amount of time to recognize that the solution is optimal. Round-off error can also compound in a very large problem as matrix inversion is repeated through several iterations.

A program listing for Kao's method coded in FORTRAN IV is found in Appendix B. An example problem is solved in Section 5.

4. THE OPTIMAL STATE-AGE REPLACEMENT POLICY ALGORITHM

Prior to the development of an algorithm to find an optimal state-age replacement policy, an equation for the expect long term average cost per unit time, $\psi(\underline{\kappa})$, must be derived. This cost is a function of the expected mean sojourn time in each state, the costs of maintenance and replacement in each state, and the associated probabilities of deterioration and replacement among the states.

Let $\{Z_t, t \geq 0\}$ be a semi-Markov process with state space E as previously defined, where L denostes the failed state. Now let $\{X_n, n = 0, 1, \ldots, L\}$ be the imbedded Markov chain associated with the semi-Markov process Z using decision $\underline{\kappa}$. Let $\hat{P}$ denote its transition matrix ($\hat{P}$ actually depends on κ but for notational convenience this dependence will not be shown). Then $\hat{P}$ is defined by

$$\hat{P}(i,j) = q(i,j,\underline{\kappa}) \text{ for } i,j \in E \text{ under the policy } \underline{\kappa}.$$

Let Δ be a state not in E and let $P(i,\Delta) = 1 - \sum_{j \in E} P(i,j)$.
Thus Δ represents the "death" of the Markov chain X.

Define the random variable ζ by

$$\zeta = \inf\{n : X_n \in L\} \wedge \sup\{n : X_n \in E\}.$$

Now the expected cost of replacement may be defined as

$$E_i[c(X_\zeta)] = c(L)\hat{P}(i,L) + c(i)\hat{P}(i,\Delta) + \sum_{k \in E\setminus\{L\}} \hat{P}(i,k)E_k[c(X_\zeta)]$$

$$= \sum_{k \in E \setminus \{L\}} R(i,k)[c(L)\hat{P}(k,L) + c(k)\hat{P}(k,\Delta)].$$

Since

$$R(i,j) = \sum_{n=0}^{\infty} \hat{P}^n(i,j), \text{ and } \hat{P}(L,L) = 0$$

then

$$R\,\hat{P}(i,j) = R(i,j) - I(i,j)$$

and thus

$$E_i[c(X_\zeta)] = \sum_{k \in E} R(i,j)\; c(k)\; \hat{P}(k,\Delta).$$

It should be noted that

$$\hat{P}(i,\Delta) = 1 - \hat{P}(i,E),$$

$$\hat{P}(L,j) = 0 \text{ for } j \quad E,$$

and $\quad \hat{P}(L,\Delta) = 1.$

It is now necessary to find the expected cycle time $E_i[\zeta]$. The expected sojourn time in state i can be written as

$$m(i) = \int_0^{\kappa_i} [1 - Q(i,E,t)]\, dt.$$

Recalling that the mean replacement time at state i is $\bar{\tau}_i$, it follows that

$$E_i[\zeta] = m(i) + \bar{\tau}_L\; \hat{P}(i,L) + \bar{\tau}_i\; \hat{P}(i,\Delta) + \sum_{k \in E \setminus \{L\}} \hat{P}(i,k)\; E_k[\tau]$$

$$= \sum_{k \in E \setminus \{L\}} R(i,k)[m(k) + \bar{\tau}_L\; \hat{P}(k,L) + \bar{\tau}_k\; \hat{P}(k,\Delta)]$$

$$= R(i,L)\; \bar{\tau}_L + \sum_{k \in E \setminus \{L\}} R(i,k)\; m(k) + \bar{\tau}_k\; \hat{P}(k,\Delta)$$

and $\quad E_i[\zeta] = \sum_{k \in E} R(i,k)\,[m(k) + \bar{\tau}_k\,\hat{P}(k,\Delta)]$ where $m(L) = 0$.

The Markov renewal kernel $R(i,j)$ may now be shown to be $R(i,j) = [I - \hat{P}]^{-1}$ for a given policy $\underline{\kappa}$. The inverse of this upper right triangular matrix may be easily computed as follows:

$$r(i,0) = 0$$

$$r(i,i) = 1$$

$$r(i,i+1) = r(i,i)\; q(i,i+1,\kappa_i)$$

$$r(i,i+2) = r(i,i)\; q(i,i+2,\kappa_i) + r(i,i+1)\; q(i+1,i+2,\kappa_{i+1})$$

$$\vdots$$

Since the system will always start a cycle in state 0 and return to state 0 upon replacement, the only elements of the Markov renewal matrix of concern to this problem are $r(0,j)$ for all $j \in E$.

The expected long term average cost per unit time can now be defined as the total expected replacement and maintenance costs per cycle divided by the expected cycle time, or

$$\psi(\underline{\kappa}) = \frac{\sum_{i=0}^{L} r(0,i)\,[m(i)\,a_i + c_i\,P(i,0)]}{\sum_{i=0}^{L} r(0,i)\,[m(i) + \bar{\tau}_i\,P(i,0)]}$$

where $c_i = b_i + \bar{\tau}_i d_i$ and $P(i,0) = P(i,\Delta)$. This expression is valid for any feasible policy κ; all that now remains is the development of an efficient algorithm to find the optimal state-age replacement policy.

An obvious initial policy for beginning a search for the optimal is to replace the system only when it fails. This may be accomplished by setting κ_i equal to its maximum value for $i = 0, 1, \ldots, L-1$ and $\kappa_L = 0$. Evaluation of this policy gives the expected long term average cost per unit time for the basic problem without early replacement. Starting in state L-1, the decision at each state may be varied to find the minimum cost, all other decisions remaining constant. By passing through the states from L-1 to 0 and adopting the minimum cost decision κ_i as part of the new policy as the search passes through state i, the resulting policy will be at least as good as the old policy, and it will be better if the old policy was not the optimal. It is not necessary to evaluate every possible κ_i at each state i. Since the costs are non-decreasing in i and the cumulative probability distribution of sojourn times is also non-decreasing in i, $\kappa_i \geq \kappa_{i+1}$. This is a logical conclusion in that it could not be of economic advantage to remain in a more expensive state longer than the limiting time of a less expensive state. Therefore, in varying the decision at state i, evaluations are only made between κ_{i+1} and the upper bound of κ_i as determined by the holding time mass function for the first iteration.

A single pass may not yield the optimal solution; since all states are initially set at the maximum decision value, a bias may exist in the decisions computed for the higher numbered states. This bias may be removed by successive passes through the states, noting that κ_j may only decrease for subsequent passes through

state j, $j \in E$. Thus the search space is decreased somewhat with each successive iteration, as the maximum κ_j is the old policy κ_j and the minimum is the current policy κ_{j+1}. This non-increasing property of decisions in successive iterations is again due to the non-decreasing properties of the system parameters. The only cause for a change in κ_j is a subsequent new minimum found at κ_i where $i < j$. Since the new κ_i will be smaller than the old κ_i, its only possible influence on κ_j will be to make it smaller than the previous decision. If the state by state iterative passes are continued until one complete pass is made with no change in the overall policy, the optimal state-age replacement policy has been found.

It must be noted at this point that two assumptions have been made and used in development of the algorithm with no proof as to their validity. The first is that the decision κ_i will be non-increasing as i increases. The other is that the decision κ_i will be non-increasing through successive iterations. Both assumptions are based on the author's intuition given the increasing nature of the costs as the state increases and the increasing likelihood of failure as the state and sojourn times increase. They have been shown to be true in the solution of the sample problems to be referenced in Sections 5 and 6. A formal proof of these properties is beyond the scope of this thesis.

A listing of the Optimal State-Age Replacement Policy Algorithm coded in FORTRAN IV may be found in Appendix C. An example problem is solved in Section 5 using the algorithm.

5. AN EXAMPLE PROBLEM

To illustrate the solution procedures for Kao's State-Age Dependent Policy and the Optimal State-Age Replacement Policy presented in Section 4, an example is given with the following parameters:

The state space E is {0, 1, 2, 3}.

The replacement costs are:

$c_0 = 2.0 \quad (b_0 = 1.0,\ d_0 = 1.0,\ \bar{\tau}_0 = 1.0)$

$c_1 = 2.0 \quad (b_1 = 1.0,\ d_1 = 1.0,\ \bar{\tau}_1 = 1.0)$

$c_2 = 2.0 \quad (b_2 = 1.0,\ d_2 = 1.0,\ \bar{\tau}_2 = 1.0)$

$c_3 = 6.0 \quad (b_3 = 2.0,\ d_3 = 2.0,\ \bar{\tau}_3 = 2.0)$.

The transition probability matrix P is

$$P = \begin{bmatrix} 0.0 & 0.6 & 0.2 & 0.2 \\ 0.0 & 0.0 & 0.7 & 0.3 \\ 0.0 & 0.0 & 0.0 & 1.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$$

The state holding time probabilities, $h(i,t)$, are:

$h(0,0) = 0.0$, $h(0,1) = 0.1$, $h(0,2) = 0.18$, $h(0,3) = 0.216$,

$h(0,4) = 0.454$, $h(0,5) = 0.050$, $H(0,5) = 1.0$,

$h(1,0) = 0.0$, $h(1,1) = 0.2$, $h(1,2) = 0.14$, $h(1,3) = 0.528$,

$h(1,4) = 0.119$, $h(1,5) = 0.013$, $H(1,5) = 1.0$,

$h(2,0) = 0.0$, $h(2,1) = 0.3$, $h(2,2) = 0.28$, $h(2,3) = 0.378$,

$h(2,4) = 0.040$, $h(2,5) = 0.002$, $H(2,5) = 1.0$,

$h(3,0) = 0.0$, with $h(3,j)$ undefined for $j \neq 0$.

Kao's Method

Prior to beginning the Policy Iteration step of Kao's method, the modified transition probabilities, mean holding times, mean waiting times, and expected cost rates must be calculated. These are tabulated below in Table 1.

Table 1. Data used for State-Age Dependent Policy (Kao's Method)

State	Alt	Transition Probabilities				Mean Holding Times		Mean Wait Time	Exp. Cost Rate
i	κ_i	p_{i0}	p_{i1}	p_{i2}	p_{i3}	$\bar{\tau}_{i0}$	$\bar{\tau}_{ij}$	$\tau(i,\kappa_i)$	$f(i,\kappa_i)$
0	0	1.000	0.000	0.000	0.000	1.0	0.0000	1.000	2.0000
	1	0.900	0.060	0.020	0.020	2.0	1.0000	1.900	1.4737
	2	0.720	0.168	0.056	0.056	3.0	0.6429	2.620	0.2748
	3	0.504	0.298	0.099	0.099	4.0	2.2339	3.124	1.1613
	4	0.050	0.570	0.190	0.190	5.0	3.0779	3.174	1.0158
	5	0.000	0.600	0.200	0.200	6.0	3.1740	3.174	1.0000
1	0	1.000	0.000	0.000	0.000	1.0	0.0000	1.000	2.0000
	1	0.800	0.000	0.140	0.060	2.0	1.0000	1.800	1.4444
	2	0.660	0.000	0.238	0.102	3.0	1.4118	2.460	1.2683
	3	0.132	0.000	0.608	0.260	4.0	2.3779	2.592	1.0509
	4	0.013	0.000	0.691	0.296	5.0	2.5735	2.605	1.0050
	5	0.000	0.000	0.700	0.300	6.0	2.6050	2.605	1.0000
2	0	1.000	0.000	0.000	0.000	1.0	0.0000	1.000	2.0000
	1	0.700	0.000	0.000	0.300	2.0	1.0000	1.700	1.4118
	2	0.420	0.000	0.000	0.580	3.0	1.4828	2.120	1.1981
	3	0.042	0.000	0.000	0.958	4.0	2.0814	2.162	1.0194
	4	0.002	0.000	0.000	0.998	5.0	2.1583	2.164	1.0009
	5	0.000	0.000	0.000	1.000	6.0	2.1640	2.164	1.0000
3	0	1.000	0.000	0.000	0.000	2.0	0.0000	2.000	3.0000

Selecting the decisions with the minimum expected cost rates, the initial policy $\underline{\kappa}$ is (5,5,5,0). The gain equations will be:

$$v(0,\underline{\kappa}) + 3.174\ g(\underline{\kappa}) = 3.174 + 0.0\ v(0,\underline{\kappa}) + 0.6\ v(1,\underline{\kappa}) + 0.2\ v(2,\underline{\kappa}) + 0.2\ v(3,\underline{\kappa})$$

$$v(1,\underline{\kappa}) + 2.605\ g(\underline{\kappa}) = 2.605 + 0.0\ v(0,\underline{\kappa}) + 0.0\ v(1,\underline{\kappa}) + 0.7\ v(2,\underline{\kappa}) + 0.3\ v(3,\underline{\kappa})$$

$$v(2,\underline{\kappa}) + 2.164\ g(\underline{\kappa}) = 2.164 + 0.0\ v(0,\underline{\kappa}) + 0.0\ v(1,\underline{\kappa}) + 0.0\ v(2,\underline{\kappa}) + 1.0\ v(3,\underline{\kappa})$$

$$v(3,\underline{\kappa}) + 2.000\ g(\underline{\kappa}) = 6.000 + 1.0\ v(0,\underline{\kappa}) + 0.0\ v(1,\underline{\kappa}) + 0.0\ v(2,\underline{\kappa}) + 0.0\ v(3,\underline{\kappa})$$

Setting $v(3,\underline{\kappa}) = 0.0$, the gain equations may be solved to yield

$$g(\underline{\kappa}) = 1.49513$$

$$v(0,\underline{\kappa}) = -3.00974$$

$$v(1,\underline{\kappa}) = -2.03984$$

$$v(2,\underline{\kappa}) = -1.07146$$

$$v(3,\underline{\kappa}) = 0.0$$

Solution of the Γ equations yields a new policy of $\underline{\kappa} = (3,0,0,0)$. Four iterations are required to find the optimal policy, $\underline{\kappa}^* = (4,2,0,0)$ with expected long term average cost rate of $\psi(\underline{\kappa}^*) = 0.32315$. A tabulation of the iterations may be found in Table 2. Execution of the FORTRAN program for this method required 0.0671 seconds under a FORTRAN IV-H compiler.

The Optimal State-Age Algorithm

The same problem can be solved using the Optimal State-Age Replacement Algorithm in only two iterations. The initial policy is the same as that of Kao's method, $\underline{\kappa} = (5,5,5,0)$. Step-wise calculations are provided in Table 3. The solution is identical to Kao's method, with

$\underline{\kappa}^* = (4,2,0,0)$ and $\psi(\underline{\kappa}^*) = 1.32315$. Execution of the FORTRAN program took only 0.0232 seconds, a 2.9 to 1 advantage over Kao's method.

Table 2. Results from State-Age Dependent Policy (Kao's Method)

State	Alt.	Iteration 1	Iteration 2	Iteration 3	Iteration 4
i	κ_i	Γ_i	Γ_i	Γ_i	Γ_i
0	0	2.0000	2.0000	2.0000	2.0000
	1	1.5564	1.5358	1.5276	1.5277
	2	1.4428	1.4011	1.3843	1.3846
	3	1.4109*	1.3489*	1.3241	1.3243
	4	1.4861	1.3693	1.3225*	1.3232*
	5	1.4951	1.3722	1.3229	1.3236
1	0	1.0301*	1.3489	1.6344	1.6318
	1	1.1567	1.2434	1.4057	1.4043
	2	1.1863	1.2035*	1.3421*	1.3232*
	3	1.4335	1.2841	1.4051	1.4045
	4	1.4888	1.3031	1.4249	1.4244
	5	1.4951	1.3053	1.4273	1.4268
2	0	0.0618*	1.3489*	1.3241*	1.3232*
	1	0.8027	1.6115	1.6057	1.6054
	2	1.1073	1.7944	1.7963	1.7964
	3	1.4566	2.1815	2.1920	2.1924
	4	1.4933	2.2229	2.2344	2.2348
	5	1.4951	2.2251	2.2366	2.2370
3	0	1.4951*	1.3489*	1.3241*	1.3232*
Policy Evaluation					
$g(\underline{\kappa})$		1.49513	1.34889	1.32406	1.32315
$v(0,\underline{\kappa})$		-3.00974	-3.30220	-3.35187	-3.35368
$v(1,\underline{\kappa})$		-2.03984	-2.65109	-2.98630	-2.98547
$v(2,\underline{\kappa})$		-1.07146	-2.65109	-2.67593	-2.67683
$v(3,\underline{\kappa})$		0.0	0.0	0.0	0.0

$\underline{\kappa}^* = (4,2,0,0)$

$\psi(\underline{\kappa}^*) = 1.32315$

Table 3. Results of Optimal State-Age Replacement Policy Algorithm

Iter. #	State i	Decision κ_i	Renewal Kernel $r(0,0)$	$r(0,1)$	$r(0,2)$	$r(0,3)$	Cost Rate $\psi(\underline{\kappa})$
1	2	0	1.0000	0.6000	0.6200	0.3800	1.34984*
		1	1.0000	0.6000	0.6200	0.5660	1.38972
		2	1.0000	0.6000	0.6200	0.7396	1.42743
		3	1.0000	0.6000	0.6200	0.9740	1.48869
		4	1.0000	0.6000	0.6200	0.9988	1.49482
		5	1.0000	0.6000	0.6200	1.0000	1.49513
	1	0	1.0000	0.6000	0.2000	0.2000	1.36580
		1	1.0000	0.6000	0.2840	0.2360	1.34092
		2	1.0000	0.6000	0.3428	0.2612	1.32340*
		3	1.0000	0.6000	0.5646	0.3562	1.34443
		4	1.0000	0.6000	0.6145	0.3777	1.34928
		5	1.0000	0.6000	0.6200	0.3800	1.34984
	0	2	1.0000	0.1680	0.0960	0.0731	1.37228
		3	1.0000	0.2976	0.1700	0.1296	1.32406
		4	1.0000	0.5700	0.3257	0.2481	1.32315*
		5	1.0000	0.6000	0.3428	0.2612	1.32340
2	1	0	1.0000	0.5700	0.1900	0.1900	1.36393
		1	1.0000	0.5700	0.2699	0.2242	1.34008
		2	1.0000	0.5700	0.3257	0.2481	1.32315*
	0	2	1.0000	0.1680	0.0960	0.0731	1.37228
		3	1.0000	0.2976	0.1700	0.1296	1.32406
		4	1.0000	0.5700	0.3257	0.2481	1.32315**

$\underline{\kappa} = (4,2,0,0)$

$\psi(\underline{\kappa}) = 1.32315$

6. A COMPARISON OF THE METHODS

In order to compare the Optimal State-Age Algorithm with Kao's method, both solution procedures have been programmed in FORTRAN IV. A main program was developed to do all data input operations, preliminary calculations required by both methods, and checks to insure that the problem parameters are acceptable. The main program then calls a subroutine for Kao's method, recording the execution time of the subroutine; it then calls a subroutine for the Optimal State-Age Algorithm and again records the execution time. Output for the problem solution is provided by the individual subroutines. The execution times are printed by the main program prior to termination. Thus an unbiased comparison between the methods is provided based only on the separate solution procedures unhampered by the calculations that are common to both methods.

The results of forty-five sample problems compared by this computer program are contained in Table 4. Some of the problems were read in on data cards while others were generated by a subroutine employing a random number generator. This exercise has shown the Optimal State-Age Algorithm to be superior to Kao's method. A ratio of execution times was computed as a measure of the computational efficiency, with Kao's method execution time in the denominator and the Optimal State-Age Algorithm in the numerator. The smallest ratio was 1.61 for a four state, six discrete time unit problem. The largest ratio was 18.70 for a thirty state, twenty discrete time unit problem. A statistical analysis was not performed on these results, as inspection of the ratios clearly implied that the ratio

is a function of the simplicity of the solution as well as the problem size. This is caused by the increase in the number of iterations that are required by Kao's method for a difficult problem. The Optimal State-Age Algorithm required a maximum of three iterations for the largest ratio problem, while Kao's method required eleven iterations. In general it appears that the new method will be in the neighborhood of two or more times as efficient as Kao's method. The Optimal State-Age subroutine also compiles faster than Kao's method's subroutine.

An additional benefit of the Optimal State-Age Algorithm is a reduction in the required array space required by the program. This array space and the actual coding will vary with the size of the problem under consideration, but it will be substantially less with the Optimal State-Age Algorithm. For a thirty state, twenty discrete time unit problem, core requirements for main program and Optimal State-Age subroutine excluding compiler and input software are 9606 bytes. The same problem requires 169,372 bytes of core for Kao's method. Although this space savings may not be of the same magnitude in smaller problems, it does allow the Optimal State-Age Algorithm to be employed on computers with much smaller storage areas than would be possible with Kao's method.

It can now be concluded that a more efficient method has been found in the Optimal State-Age Algorithm. Thus the objective of this research has been achieved. Although the method is restricted to a special class of problems as stated in Section 2, the concept of a systematic search using a closed form cost expression has been shown to be both feasible and efficient. Further research in this area to develop an algorithm for a more general model is recommended.

Table 4. Comparison of Execution Times

Number of States	Range of i	Execution Time in Seconds A Kao's Method	Execution Time in Seconds B Optimal State-Age Algorithm	Ratio A/B
4	6	0.0122	0.0076	1.61
4	6	0.0161	0.0089	1.81
4	6	0.0191	0.0091	2.10
4	10	0.0151	0.0082	1.84
4	10	0.0215	0.0110	1.95
4	10	0.0201	0.0110	1.83
4	20	0.0276	0.0124	2.23
4	20	0.0327	0.0142	2.30
4	20	0.0302	0.0140	2.16
8	6	0.0462	0.0145	3.19
8	6	0.0460	0.0204	2.25
8	6	0.0489	0.0187	2.61
8	10	0.0606	0.0176	3.44
8	10	0.0601	0.0274	2.19
8	10	0.0623	0.0255	2.44
8	20	0.1327	0.0269	4.93
8	20	0.1182	0.0461	2.56
8	20	0.1212	0.0472	2.57
10	6	0.0884	0.0187	4.73
10	6	0.0892	0.0301	2.96
10	6	0.0748	0.0296	2.53
10	10	0.1127	0.0248	4.54
10	10	0.0966	0.0421	2.29
10	10	0.0971	0.0414	2.35
10	20	0.2118	0.0398	5.32
10	20	0.1908	0.0734	2.60
10	20	0.1091	0.0477	2.28
20	6	0.4895	0.0665	7.36
20	6	0.5992	0.1321	4.54
20	6	0.3953	0.1487	2.66
20	10	0.6999	0.0922	7.59
20	10	0.5919	0.1940	3.05
20	10	0.5954	0.2171	2.74
20	20	1.1532	0.1612	7.15
20	20	0.9020	0.3624	2.49
20	20	0.9111	0.1183	7.70
30	6	1.4567	0.1704	8.55
30	6	1.7972	0.3578	5.02
30	6	1.4970	0.4750	3.15
30	10	1.6848	0.3051	5.52
30	10	1.7010	0.5466	3.11
30	10	1.7292	0.7362	2.35
30	20	5.0424	0.4335	11.63
30	20	3.5923	0.4257	8.44
30	20	3.5961	0.1923	18.70

REFERENCES

BARLOW, R. E., PROSCHAN, F., and HUNTER, L. C. (1965) The Mathematical Theory of Reliability. Wiley, New York.

CINLAR, E. (1975) Introduction to Stochastic Processes. Prentice-Hall, Englewood Cliffs, N. J.

DERMAN, C. (1962) On sequential decision and Markov chains, Management Science 9, 16-24.

FELDMAN, R. M. (1976) Optimal replacement with semi-Markov shock models. J. Appl. Probability 13, 108-117.

HOWARD, R. (1960) Dynamic Programming and Markov Processes. The MIT Press, Cambridge, Mass.

KAO, E. P. C. (1972) Optimal replace rules when changes of state are semi-Markovian. Oper. Res. 21, 1231-1249.

LAMPE, M. E. (1977) A comparison of methods for finding the optimal replacement rule of a semi-Markov replacement process, A Research Report, Department of Industrial Engineering, Texas A&M University, College Station, Texas

ROSS, S. M. (1970) Applied Probability Models with Optimization Applications. Holden-Day, San Francisco.

TAYLOR, H. M. (1975) Optimal replacement under additive damage and other failure models. Naval Res. Logist. Quart. 22, 1-18.

APPENDIX A

MAIN PROGRAM

Following is a program listing for the main program used to evaluate the efficiency of Kao's State-Age Dependent Policy and the Optimal State-Age Replacement Policy Algorithm developed in this paper. The program is written in FORTRAN IV for a H extended compiler and should easily adapt to any machine with a FORTRAN IV compiler. The Subroutine TIMER is a system subroutine unique to the Texas A&M University Data Processing System; its function is to keep track of actual execution time for each of the method subroutines and it can be easily replaced by a routine to access the computer's internal clock prior to calling and upon return from the method subroutines. This listing is dimensioned for a thirty state, twenty possible discrete time problem. For larger problems or to save array space in smaller problems, the user need only change the DIMENSION and COMMON statements to reflect the problem size desired.

```
C     ******************************************************************************
C
C     MAIN PROGRAM... SETS UP DATA (GENERATED OR READ IN) AND CALLS HPIM
C     AND/OR MFM FOR SOLUTION
C
C     ******************************************************************************
C   DEFINE VARIABLES
      COMMON P(30,30),H(30,20),A(30),R(30),D(30),TR(30),PK(30,30,20),
     1KK(30),K(30),LM1,LP1,L,IM
      DIMENSION KN(30,30), RR(20)
      DIMENSION HH(20)
C   INPUT
12345 CONTINUE
      READ(5,100) L,IM
  100 FORMAT(2I5)
      IF(L.EQ.0) GO TO 5555
      LM1 = L - 1
      LP1 = L + 1
C   CALL INPUT SUBROUTINE TO READ IN DATA, GENERATE DATA, OR A COMBINATION
C   OF BOTH.  MAXIMUM PROBLEM SIZE IS DETERMINED BY ARRAY DIMENSIONS.
      CALL INPUT
C   SET LIMIT ON EACH K(I)
      DO 90 I = 1,L
      DO 91 IK = 2,IM
      IF(H(I,IK)) 92,92,91
   92 KK(I) = IK - 1
      GO TO 90
   91 CONTINUE
      KK(I) = IM
   90 CONTINUE
C   CALCULATE TRANSITION PROBABILITIES
      DO 1 I = 1,L
      DO 1 J = 2,L
      MM = KK(I)
      DO 1 IK = 1,MM
      SUM = 0.0
```

```
      IF(P(I,J) .LE. 0.0) GO TO 1
      MN = IK
      DO 2 KI = 1, MN
    2 SUM = SUM + H(I,KI)
    1 PK(I,J,IK) = P(I,J) * SUM
      DO 3 I = 1,LM1
      MM = KK(I)
      DO 3 IK = 1,MM
      SUM1 = 0.0
      IP1 = I + 1
      DO 4 J = IP1,L
      SUM = 0.0
      IKP1 = IK + 1
      MN = KK(I)
      IF(IKP1 .GT. MN) GO TO 512
      DO 5 KI = IKP1,MN
    5 SUM = SUM + H(I,KI)
  512 CONTINUE
    4 SUM1 = SUM1 + P(I,J) * SUM
    3 PK(I,1,IK) = SUM1
      MM = KK(I)
      DO 6 IK = 1,MM
    6 PK(L,1,IK) = 1.0
      DO 7 I = 1,L
      DO 8 J = 2,L
    8 PK(I,J,1) = 0.0
    7 PK(I,1,1) = 1.0
      WRITE(6,250)
      DO 800 I = 1,L
C     ECHO CHECK ON INPUT DATA
  800 WRITE(6,251) (P(I,J),J = 1,L)
      WRITE(6,252)
      DO 801 I = 1,L
      IX = I - 1
      KX = KK(I)
      WRITE(6,253) IX,(H(I,II),II = 1,KX)
```

```
  801 CONTINUE
C     INTERNAL CHECK THAT PROBABILITIES ALL SUM TO 1.00
      DO 803 I = 1,L
      S = 0.0
      DO 802 J = 1,L
  802 S = S + P(I,J)
      ISS = S + .00005
      S = ISS
  803 IF(S . NE. 1.0) WRITE(6,255) I,S
      DO 805 I = 1,LM1
      S = 0.0
      DO 804 J = 1,IM
  804 S = S + H(I,J)
      ISS = S + .001
      S = ISS
  805 IF(S .NE. 1.0) WRITE(6,256) I,S
  255 FORMAT(1H0,'**** INTERNAL WARNING *** P(',I2,',(.) ) = ',F10.7)
  256 FORMAT(1H0,'**** INTERNAL WARNING *** H(',I2,',(.) ) = ',F10.7)
      WRITE(6,254)( A(I),I = 1,L)
      WRITE(6,254)( R(I),I = 1,L)
      WRITE(6,254)( D(I),I = 1,L)
      WRITE(6,254)(TR(I),I = 1,L)
  250 FORMAT(1H1,'TRANSITION PROBABILITY MATRIX'/)
  251 FORMAT(1H ,20F6.3)
  252 FORMAT(1H0,' STATE HOLDING TIME PROBABILITIES'/)
  253 FORMAT(1H ,'I = ',I3,(1X,15F7.3))
  254 FORMAT(1H0,20F6.1)
C     SET TIMER AND CALL SUBROUTINE TO EXECUTE KAO'S METHOD
      CALL TIMER(ITIME)
      CALL MPIM
      CALL TIMER(ITIME)
C     SET TIMER AND CALL SUBROUTINE TO EXECUTE NEW METHOD
      CALL TIMER(JTIME)
      CALL MFM
      CALL TIMER(JTIME)
C     NOTE..RESULTS OF EACH METHOD ARE PRINTED BY THE INDIVIDUAL SUBROUTINES
```

```
C    CALCULATE AND PRINT EXECUTION TIME OF EACH METHOD
       YTIME = ITIME / 38400.
       XTIME = JTIME / 38400.
       WRITE(6,888) YTIME,XTIME
  888 FORMAT(1H0,'HPIM TIME = ', F8.4,'SEC, MFM TIME = ',F8.4,'SEC')
       GO TO 12345
 5555 CONTINUE
       STOP
       END
```

APPENDIX B

SUBROUTINE HPIM

Following is a program listing of Subroutine HPIM, a routine to solve for the optimal state-age replacement policy for a semi-Markov deterioration process. The solution procedure is that of Kao (1972) using Howard's Policy Iteration Method. Solution of the L+1 linear equations is done by inversion in place. This listing is for a maximum size problem of thirty states and twenty possible discrete holding times per state. For larger problems or to save array space in smaller problems, the DIMENSION and COMMON statements may be modified without effecting execution of the subroutine. All output is internally coded in the subroutine so that it may be used without the main program found in Appendix A if the user supplies a program to furnish input variables and call the subroutine.

```
      SUBROUTINE HPIM
C     ************************************************************************
C
C     SUBROUTINE FOR OPTIMAL REPLACEMENT POLICY OF A SEMI-MARKOV DETERIORATION
C     PROCESS USING KAO'S METHOD WITH POLICY ITERATION
C
C     ************************************************************************
      COMMON P(30,30),H(30,20),A(30),R(30),D(30),TR(30),PK(30,30,20),
     1KK(30),K(30),LM1,LP1,L,IM
      DIMENSION TK(30,30,20),QK(30,20),TM(30,20),V(30),X(32,31),XX(32,31
     1),G(30,20),KSAVE(30),AL(30,30,20)
      GSAV = 0.
      IT = 0
C     CALCULATE MEAN SOJOURN TIMES
      DO 9 I = 1,L
      DO 9 J = 1,L
      MM = KK(I)
      DO 9 IK = 1,MM
      IF(J .LE. I) GO TO 13
      SUM = 0.0
      SUM1 = 0.0
      DO 10 KI = 1,IK
      SUM = SUM + (KI - 1) * H(I,KI)
   10 SUM1 = SUM1 + H(I,KI)
      IF(SUM1) 13,13,9
   13 SUM = 0.0
      SUM1 = 1.0
    9 TK(I,J,IK) = SUM / SUM1
      DO 11 I = 1,LM1
      MM = KK(I)
      DO 11 IK = 1,MM
   11 TK(I,1,IK) = TR(I) + IK - 1.0
      MM = KK(L)
      DO 12 IK = 1, MM
   12 TK(L,1,IK) = TR(L)
C     CALCULATE MEAN WAITING TIMES
```

```
      DO 14 I = 1,L
      MM = KK(I)
      DO 14 IK = 1,MM
      SUM = 0.0
      DO 15 J = 1,L
   15 SUM = SUM + PK(I,J,IK) * TK(I,J,IK)
   14 TM(I,IK) = SUM
C     CALCULATE EXPECTED COST RATES
      DO 16 I = 1,LM1
      MM = KK(I)
      DO 16 IK = 1,MM
      SUM = 0.0
      DO 17 J = 1,L
   17 SUM = PK(I,J,IK) * TK(I,J,IK) + SUM
      DO 16 J = 1,L
   16 AL(I,J,IK) = PK(I,J,IK) * TK(I,J,IK) / SUM
      DO 18 I = 1,LM1
      MM = KK(I)
      DO 18 IK = 1,MM
      SUM = 0.0
      IP1 = I + 1
      DO 19 J = IP1,L
   19 SUM = SUM + AL(I,J,IK) * A(I)
      DEN = TR(I) + IK - 1.0
   18 QK(I,IK)=(AL(I,1,IK)*((IK-1.0)*A(I)+D(I)*TR(I)+R(I))/DEN) + SUM
      MM = KK(L)
      DO99 IK = 1,MM
   99 QK(L,IK) = (D(L)*TR(L) + R(L))/TR(L)
C     SELECT INITIAL POLICY
      DO 20 I = 1, L
      XMIN = QK(I,1)
      K(I) = 1
      MM = KK(I)
      DO 20 KI = 1,MM
      IF(XMIN - QK(I,KI))20,20,21
   21 XMIN = QK(I,KI)
```

```
      K(I) = KI
   20 CONTINUE
  222 CONTINUE
C     POLICY EVALUATION STEP
      IT = IT + 1
      DO 31 I = 1,L
      X(I,L+1) = QK(I,K(I))*TM(I,K(I))
      X(I,L) = TM(I,K(I))
      DO 32 J = 1, LM1
   32 X(I,J) = PK(I,J,K(I))* (-1.0)
   31 X(I,I) = X(I,I) + 1.0
      X(L,L) = X(L,L) - 1.0
C     SOLUTION OF L SIMULTANEOUS LINEAR EQUATIONS BY INVERSION IN PLACE
      DO 45 I = 1,L
      XX(I,I) = 1.0
      DO 42 J = 1,LP1
   42 XX(I,J) = X(I,J)/X(I,I)
      DO 43 J = 1,L
   43 XX(J,I) = -1.0 * X(J,I)/X(I,I)
      DO 41 N = 1,L
      DO 41 M = 1,LP1
      IF(N.EQ.I) GO TO 41
      IF(M.EQ.I) GO TO 41
      XX(N,M) = X(N,M) - X(N,I) * X(I,M) / X(I,I)
   41 CONTINUE
      DO 44 II = 1,L
      DO 44 JJ = 1,LP1
      X(II,JJ) = XX(II,JJ)
   44 CONTINUE
   45 CONTINUE
      DO 33 I = 1,LM1
   33 V(I) = XX(I,L+1)
      V(L) = 0.0
      GG= XX(L,L+1)
C     POLICY IMPROVEMENT OPERATION
      DO 50 I = 1,L
```

```
      MM = KK(I)
      DO 50 IK = 1,MM
      SUM = 0.0
      DO 51 J = 1,L
   51 SUM = SUM + PK(I,J,IK) * V(J)
   50 G(I,IK) = QK(I,IK) +(SUM-V(I))/TM(I,IK)
      DO 53 I = 1,L
      XMIN = G(I,1)
      KX = 1
      MM = KK(I)
      DO 52 IK = 1,MM
      IF(G( I,IK).GE.XMIN) GO TO 52
      XMIN = G(I,IK)
      KX = IK
   52 CONTINUE
      KSAVE(I) = K(I)
   53 K(I) = KX
      WRITE(6,206) IT
      WRITE(6,207) GG
  206 FORMAT(1H ,'RESULTS OF ITERATION ',I3/1X,'POLICY EVALUATION OPERAT
     1ION')
  207 FORMAT(1H ,'GAIN = ',F10.5)
C     CHECK NEW POLICY AGAINST OLD POLICY
      IF(GSAV.EQ.GG) GO TO 54
      GSAV = GG
      DO 54 I = 1,L
      IF(K(I) .NE. KSAVE(I)) GO TO 222
   54 CONTINUE
      WRITE(6,200) IT
  200 FORMAT(1H0,'OPTIMUM SOLUTION FOUND IN ',I3,' ITERATIONS')
      DO 55 I = 1,L
      J = K(I)
      M = J - 1
      II = I - 1
   55 WRITE(6,201) II,M,G(I,J)
  201 FORMAT(1H ,'K(',I2,') = ',I2,5X,' EXP. COST =',F10.5)
```

APPENDIX C

SUBROUTINE MFM

Following is a program listing of Subroutine MFM, a routine to solve for the optimal state-age replacement policy for a semi-Markov deterioration process. The solution procedure is the one developed in this paper, the Optimal State-Age Replacement Policy Algorithm, using a closed form expected long term average cost per unit time expression and a selective search technique to find the minimum cost policy. This listing is for a maximum size problem of thirty states and twenty possible discrete holding time units per state. For larger problems or to save array space in smaller problems, the user may modify the DIMENSION and COMMON statements to the appropriate size without adverse effect on the routine execution. All output is internally coded in the subroutine so that it may be used without the main program found in Appendix A if the user supplies a suitable main program to input the parameters of the problem.

```
      SUBROUTINE MFM
C    ********************************************************************************
C
C   SUBROUTINE FOR OPTIMAL REPLACEMENT POLICY FOR A SEMI-MARKOV DETERIORATION
C    PROCESS USING A CLOSED FORM EXPRESSION FOR EXPECTED AVERAGE LONG TERM
C    COST PER UNIT TIME AND A SELECTIVE SEARCH PROCEDURE FOR THE MINIMUM
C
C    ********************************************************************************
      COMMON P(30,30),H(30,20),A(30),R(30),D(30),TR(30),PK(30,30,20),
     1KK(30),K(30),LM1,LP1,L,IM
      DIMENSION PHI(20),RM(30),TM(30,20)
      DIMENSION KSAVE(30)
C    SET INITIAL POLICY
      ISTART = 1
      KSAVE(L) = 1
      K(L) = 1
      TM(L,1) = 0.0
      DO 1  J = 1, LM1
      KSAVE(J) = KK(J)
    1 K(J) = KK(J)
C    CALCULATE MEAN SOJOURN TIMES FOR INITIAL POLICY
      DO 2 J = 1, LM1
      TM(J,K(J)) = 0.0
      INDX = K(J) - 1
      DO 2 I = 1, INDX
    2 TM(J,K(J)) = TM(J,K(J)) + PK(J,1,I)
   88 IX = 2
C    BEGIN SEARCH FOR MINIMUM COST DECISION IN STATE LS, LS = L-1 TO 0 FOR
C    FIRST ITERATION, RESET UPPER BOUND FOR EACH ADDITIONAL ITERATION
      DO 3 LCC = ISTART,LM1
      LS = L - LCC
      LSP = LS + 1
      LSM = LS - 1
C    CALCULATE MARKOV RENEWAL KERNEL FOR CURRENT POLICY
      KX = KSAVE(LS)
      RM(1) = 1.0
```

```
      IA = K(LS+1)
      DO 8 I =IA,KX
      K(LS) = I
      DO 4 J = IX,L
      S = 0.0
      JX = J - 1
      DO 5 JK = 1,JX
    5 S = S +RM(JK) * PK(JK,J,K(JK))
    4 RM(J) = S
      IX = LSM
      IF(IX .LT. 2) IX = 2
      S = 0.0
      IM1 = I - 1
      IF(IM1.EQ.0) GO TO 11
C     CALCULATE MEAN SOJOURN TIME FOR CURRENT DECISION
      DO 6 J = 1,IM1
    6 S = S + PK(LS,1,J)
      TM(LS,I) = S
   11 TM(LS,1) = 0.0
C     CALCULATE EXPECTED COST FOR CURRENT POLICY
      S = 0.0
      DD = 0.0
      DO 7 J = 1,L
      S = S + RM(J) * (TM(J,K(J))*A(J) + PK(J,1,K(J))*(R(J)+D(J)*TR(J)))
    7 DD=DD + RM(J)*(TM(J,K(J)) + PK(J,1,K(J))*TR(J))
    8 PHI(I) = S / DD
C     SELECT MINIMUM COST DECISION FOR THIS STATE
      PTST = PHI(IA)
      K(LS) = IA
      IAP = IA + 1
      IF(IAP .GT. KX) GO TO 9
      DO 9 I = IAP,KX
      IF(PHI(I) .GT. PTST) GO TO 9
      PTST = PHI(I)
      K(LS) = I
    9 CONTINUE
```

```
    3 CONTINUE
C     CHECK FOR A CHANGE OF POLICY DURING THIS ITERATION
      ITER = 0
      J = ISTART
      DO 12 LCC = ISTART,LM1
      I = L - LCC
      IF(K(I) .EQ. 1) J = I
      IF(K(I) .EQ. KSAVE(I)) GO TO 12
      KSAVE(I) = K(I)
      ITER = 1
   12 CONTINUE
C     SET UPPER SEARCH LIMIT AT HIGHEST NON-ZERO DECISION
      ISTART = J - 1
      IF(ISTART .LE. 0) ISTART = 1
C     IF POLICY CHANGED THIS ITERATION, DO AN ADDITIONAL ITERATION
      IF(ITER)56,56,88
C     WRITE OPTIMAL POLICY AND EXPECTED LONG TERM AVERAGE COST PER UNIT TIME
   56 WRITE(6,200) PTST
      DO 10 I = 1,L
      K(I) = K(I) - 1
      IX = I - 1
   10 WRITE(6,201)IX,K(I)
  200 FORMAT(1H1,'OPTIMUM EXPECTED AVERAGE COST IS ',F10.6/)
  201 FORMAT(1H ,'K(',I2,') = ',I2)
      RETURN
      END
```

VITA

Tilden Newton Mikel was born October 31, 1946 at Jacksonville, Texas, son of Edgar D. and Ernest Earl Mikel. He graduated from Thomas Jefferson High School, Dallas, Texas, in 1964 and did his undergraduate work at Texas A&M University, receiving a Bachelor of Science degree in Industrial Engineering and a Commission as a Second Lieutenant in the United States Army in 1969. He is currently a Captain in the Regular Army and is a graduate of the Infantry Officer Basic Course, the Infantry Officer Advance Course, Airborne School, Ranger School, Pathfinder School, and the Rotary Wing Officer Aviator Course. Captain Mikel's permanent mailing address is 3777 Pallos Verdas, Dallas, Texas.

The typist for this thesis was Natalie Olsson.